



FLASK INFINITY

CODE:

programação sem limite



INDICE GERAL

INTRODUCAO

- CAPITULO 1: Python(o que é, instalação, venv);
- CAPITULO 2: Flask;
- CAPITULO 3: Criando um site com Flask;
- CAPITULO 4: Associando Flask a um banco de dados;
- CAPITULO 5: Estilizando o site Flask;
- CAPITULO 6: Gerenciando um site com Flask .



CAPITULO 1

PYTHON

o que é, instalação, venv

PYTHON

O que é?

Python é uma linguagem de programação poderosa e versátil, amplamente utilizada para desenvolvimento de software, ciência de dados, inteligência artificial, automação, web e muito mais!

- **Por que Python é tão popular?**

1. Fácil de aprender: Possui uma sintaxe simples e intuitiva.
2. Código legível: É fácil de entender, mesmo para iniciantes.
3. Multipropósito: Pode ser usado para web, análise de dados, IA, automação e muito mais.
4. Bibliotecas incríveis: Tem milhares de pacotes que ajudam em qualquer tarefa.
5. Grande comunidade: Milhões de desenvolvedores colaboram e criam soluções.

- **O que posso fazer com Python?**

- ✓ Criar sites e APIs com Flask ou Django
- ✓ Automatizar tarefas repetitivas
- ✓ Criar bots e scripts inteligentes
- ✓ Desenvolver inteligência artificial e machine learning
- ✓ Construir jogos, aplicativos e muito mais!

Instalação

- **Verificar a necessidade de instalação**

Antes de instalar, verifique se já tem Python instalado no seu sistema:

- Windows: Abra o Prompt de Comando (cmd) e digite:

```
python --version
```

- Mac/Linux: Abra o Terminal e digite o mesmo comando.

Se Python já estiver instalado, ele mostrará a versão. Caso contrário, siga os próximos passos!

- **Baixando o instalador do Python**

1. Acesse o site oficial do Python:

<https://www.python.org/downloads/>

1. Escolha a versão mais recente do Python e clique em "Download".

2. Windows: Baixe o instalador .exe.

3. Mac: Baixe o arquivo .pkg.

4. Linux: Normalmente já vem instalado, mas pode ser atualizado via comando.

- **Instalando Python no Windows**

- Abra o arquivo baixado (python-XYZ.exe).
- Marque a opção "Add Python to PATH" (importantíssimo!).
- Clique em "Install Now" e aguarde a instalação.
- Após instalar, abra o Prompt de Comando (cmd) e digite:

```
python --version
```

Se aparecer algo como Python 3.X.X, a instalação foi concluída com sucesso!

- **Instalando Python no Mac**

Clique no arquivo .pkg baixado e siga as instruções do instalador.

- Após a instalação, verifique se o Python está instalado digitando no Terminal:

```
python3 --version
```

Caso não esteja instalado corretamente, você pode usar Homebrew para instalar:

```
brew install python
```

- **Instalando Python no Linux**

No Terminal, digite:

```
sudo apt update && sudo apt install python3 -y
```

- Para sistemas baseados em RedHat (Fedora, CentOS):

```
sudo dnf install python3
```

- Após a instalação, confirme com:

```
python3 --version
```

- **Instalando o gerenciador de pacotes (pip)**

O pip é usado para instalar bibliotecas do Python. Para instalá-lo, digite:

```
python -m ensurepip --default-pip
```

Se precisar atualizar, use:

```
pip install --upgrade pip
```

VENV

Venv é um módulo nativo do Python que permite criar ambientes virtuais isolados. Ele é essencial para gerenciar dependências e evitar conflitos entre projetos diferentes.

- **Por que usar venv?**

- ✓ Mantém as bibliotecas organizadas dentro de cada projeto.
- ✓ Evita conflitos entre versões de pacotes.
- ✓ Permite testar diferentes configurações sem afetar o sistema.

- **Como criar um venv?**

1. Abra o terminal ou prompt de comando.

2. Navegue até sua pasta de projeto:

```
cd /caminho/do/projeto
```

3. Crie o ambiente virtual:

```
python -m venv meu_venv
```

4. Ative o ambiente virtual:

- Windows:

```
meu_venv\Scripts\activate
```

- Mac/Linux:

```
source meu_venv/bin/activate
```

5. Instale pacotes dentro do venv:

```
pip install flask
```



CAPITULO 2

FLASK

O que é Flask? Como o Flask é usado?

Popularidade do Flask

FLASK

O que é?

O Flask é um framework web minimalista e poderoso para Python, usado para criar aplicações web e APIs de maneira rápida e flexível. Ele é chamado de "microframework" porque não impõe estruturas rígidas ao desenvolvedor, permitindo a criação de projetos personalizados.

Criado por Armin Ronacher em 2010, o Flask se destaca por sua simplicidade e leveza, sendo uma excelente escolha para projetos de pequeno a médio porte e até mesmo grandes aplicações, quando bem estruturado.

Como o Flask é usado?

O Flask é utilizado principalmente para desenvolver:

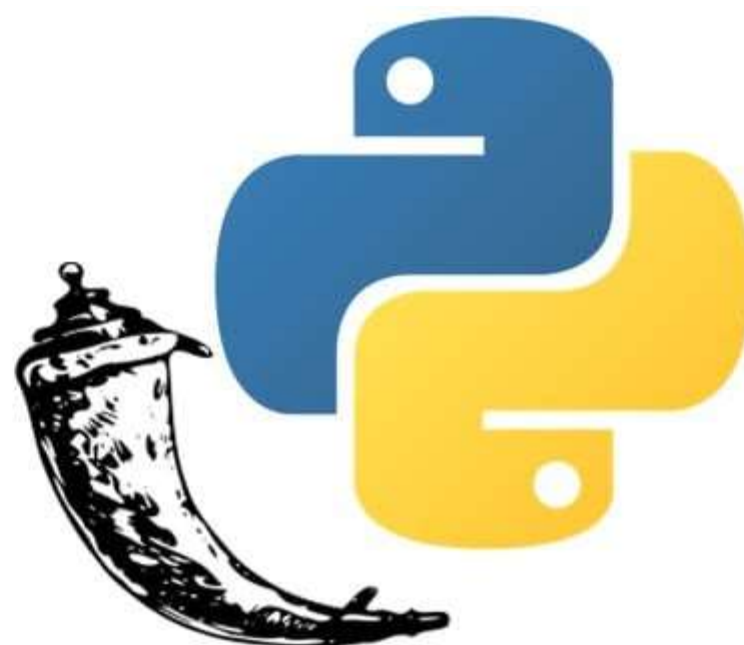
- ✓ Sites e aplicações web – Desde pequenos projetos até grandes sistemas.
- ✓ APIs RESTful – Comunicação entre front-end e back-end de forma eficiente.
- ✓ Automações – Como sistemas internos que facilitam processos empresariais.
- ✓ Integração com bancos de dados – Usando bibliotecas como SQLAlchemy para gerenciar dados.
- ✓ Desenvolvimento rápido e prototipagem – Excelente para MVPs (Minimum Viable Product).

Popularidade do Flask

O Flask é extremamente popular no mundo da programação, especialmente porque:

- ✓ É simples de aprender – Perfeito para iniciantes e empresas que precisam desenvolver rápido.
- ✓ Tem grande suporte da comunidade – Milhares de desenvolvedores contribuem com extensões e melhorias.
- ✓ Possui integração fácil com outras tecnologias – Como bancos de dados, autenticação e serviços em nuvem.
- ✓ É usado em grandes empresas – Empresas como Netflix, Uber e Reddit utilizam Flask em suas aplicações.

Embora existam outros frameworks como Django, que oferece uma estrutura mais robusta, o Flask continua sendo uma escolha favorita devido à sua flexibilidade, leveza e velocidade no desenvolvimento.



Flask

CAPITULO 3

CRIANDO UM SITE

COM FLASK

Instalação / inicialização

CRIANDO UM SITE COM FLASK

Instalação

Abra o terminal ou prompt de commando e digite:

```
pip install flask
```

Inicialização

- **Crie um arquivo app.py para começar seu projeto Flask.**

Configurando a aplicação inicial

No arquivo app.py, escreva o seguinte código:

```
Python

from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Bem-vindo ao Flask!"

if __name__ == '__main__':
    app.run(debug=True)
```

Este código inicializa o Flask e cria uma rota principal (/) que retorna uma mensagem simples.

- **Executando o servidor**

Agora, execute o seguinte comando no terminal:

```
python app.py
```

Isso iniciará um servidor local. Acesse seu site digitando `http://127.0.0.1:5000/` no navegador!



CAPITULO 3

ASSOCIANDO FLASK A UM BANCO DE DADOS

banco de dados é um sistema organizado para armazenar, gerenciar e recuperar informações de forma eficiente. Ele pode ser usado para armazenar dados de aplicativos, sites, empresas e muito mais.

ASSOCIANDO FLASK A UM BANCO DE DADOS

Instalando

Para armazenar dados, usaremos SQLite e SQLAlchemy, uma biblioteca ORM (Object-Relational Mapping).

Instalando o SQLAlchemy

```
pip install flask-sqlalchemy
```

Configurando o banco no app.py

Agora, adicione a configuração do banco ao arquivo:

Python

```
from flask_sqlalchemy import SQLAlchemy

app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///meubanco.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)
```

Criando um modelo de banco de dados

Vamos definir um modelo chamado Usuario para armazenar usuários:

Python

```
class Usuario(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    nome = db.Column(db.String(100), nullable=False)  
  
# Criar o banco de dados  
  
with app.app_context():  
    db.create_all()
```

Adicionando dados ao banco

Agora seu Flask está conectado ao banco SQLite, pronto para armazenar e gerenciar dados!

Python

```
novo_usuario = Usuario(nome="Luana")  
db.session.add(novo_usuario)  
db.session.commit()
```



CAPITULO 4

ESTILIZANDO O

SITE

HTML (HyperText Markup Language) é a linguagem de marcação usada para criar e estruturar páginas da web. Ele define elementos como textos, imagens, links e vídeos por meio de tags.

ESTILIZANDO O SITE

Escrevendo o CSS (static/style.css)

Para tornar seu site mais bonito, usaremos HTML + CSS.

Criando a pasta de estilos

Dentro do seu projeto, crie uma pasta chamada static e um arquivo style.css dentro dela.

Escrevendo o CSS (static/style.css)

Css

```
body {  
  font-family: Arial, sans-serif;  
  background-color: #f5f5f5;  
  text-align: center;  
}  
h1 {  
  color: #333;  
}
```

Criando um template HTML (templates/index.html)

Criando um template HTML (templates/index.html)

Crie uma pasta chamada templates e adicione um arquivo index.html com o seguinte código:

```
Html Copy  
  
<!DOCTYPE html>  
<html lang="pt">  
<head>  
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">  
</head>  
<body>  
    <h1>Bem-vindo ao Flask!</h1>  
</body>  
</html>
```

Alterando o app.py para renderizar o HTML

Modifique seu código para utilizar a função render_template:

```
Python  
  
from flask import render_template  
  
@app.route('/')  
def home():  
    return render_template('index.html')
```



Flask Blueprints

CAPITULO 4 GERENCIANDO SITES COM FLASK

Blueprints no Flask são uma forma de organizar e modularizar aplicações web. Eles permitem dividir grandes projetos em partes menores e reutilizáveis, facilitando a manutenção do código.

GERENCIANDO SITES COM FLASK

Organizando o código com Blueprints

Para gerenciar sua aplicação, você pode usar diversas ferramentas, como autenticação e monitoramento de erros.

Os Blueprints permitem dividir seu código em módulos organizados:

Python

```
from flask import Blueprint

admin_bp = Blueprint('admin', __name__)

@admin_bp.route('/admin')
def painel_admin():
    return "Painel administrativo"

app.register_blueprint(admin_bp)
```


Adicionando autenticação com Flask login

Adicionando autenticação com Flask Login

Para autenticar usuários, instale Flask-Login:

```
pip install flask-login
```

Agora, configure autenticação:

Python

```
from flask_login import LoginManager

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = "login"
```

Registrando logs para monitorar erros

Registrando logs para monitorar erros

Crie um sistema de logs para armazenar mensagens importantes:

Python

```
import logging

logging.basicConfig(filename='registro.log', level=logging.DEBUG)
app.logger.info("Servidor iniciado.")
```

CONCLUSÃO

Agora você tem um site completo em Flask, com banco de dados, estilização, organização e gestão eficiente!

Este projeto foi desenvolvido por: Luana Julia Gomes Vieira

Juntamente as IA's: Copilot e MidJourney

Para o Bootcamp da Dio: TONNIE - Java and AI in Europe

Desde já, Agradeço pela Atenção!